Official URL

DOI : https://doi.org/10.1002/j.2334-5837.2018.00561.x

Towards Solving MBSE Adoption Challenges: The D3 MBSE Adoption Toolbox

Mohammad Chami
Chami Consulting
Mannheim, 68259 Germany
mc@chamiconsulting.com

Aurelijus Morkevicius
No Magic Europe and Kaunas University of Technology
Kaunas, LT-51480, Lithuania
aurelijus.morkevicius@nomagic.com

Aiste Aleksandraviciene
No Magic Europe
Kaunas, LT-51480, Lithuania
aiste.aleksandraviciene@nomagic.com

Jean-Michel Bruel
IRIT, University of Toulouse
31070 Blagnac, France
bruel@irit.fr

Abstract. Increasing competition drives organizations to continually seek ways to improve the quality of their products and services, time to market and pricing pressures. Model-Based Systems Engineering (MBSE) promises many benefits to solve document-based engineering problems. However, the journey of MBSE adoption relies on several human, financial, organizational and technological factors. Organizations that decide to adopt MBSE must be aware of those factors. This paper outlines the MBSE adoption experience of a series of projects and presents an approach to support and guide organizations in overcoming MBSE adoption challenges.

Introduction

MBSE promises to facilitate communication across different engineering disciplines (Delp *et al.* 2013). To attain this goal, support of different architecture views for a single system model is required. In many cases, these views are not compatible and are completely disintegrated to each other. To deal with this challenge, the organization must implement proper practices, where language, method and tool are vital constructs.

Although MBSE benefits have been reported from different organizations and based on various experiences, few reports talk about the problems and failure stories the organizations face. Based on our experience, the discussion often centers on MBSE technical details (e.g., SysML language, modeling tool) rather than identifying and understanding the human factor, i.e., the personnel involved in the MBSE adoption and the effect they have.

It is also crucial to understand that MBSE is not a necessity for all organizations. There might be cases where MBSE adoption would generate higher costs compared to its benefits. In fact, MBSE adoption requires a holistic and systematic approach, and it differs among industry sectors and organization environments. For example, a German tool manufacturer did a study on the potential benefits of MBSE. They found their systems were too simple (one engineer can easily understand how their systems worked) to make an investment in MBSE beneficial.

Outline

In this paper, the best practices of adopting MBSE are summarized. The core components vital to successful MBSE adoption are identified and a Toolbox for successful MBSE adoption is proposed.

This paper is structured as follows: in Section 2, the background information is provided; in Section 3, the related works are analyzed; in Section 4, the D3 MBSE Adoption Toolbox is described; in Section 5, several cases of applying the proposed approach are described; in Section 6, the achieved results, conclusions, and future work directions are presented.

Background

MBSE Fundamental Concepts

According to the International Council on Systems Engineering (INCOSE 2007), MBSE is '*the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases'*. INCOSE also envisions that MBSE will become a common practice and a synonym of SE by 2020.

The MBSE movement was reinforced with the successful adoption of Unified Modeling Language (UML) (OMG 2007) and Model-Driven Architecture (MDA) (OMG 2003). However, UML was too complicated and unnatural for solving systems engineering (SE) domain-specific problems (Morkevicius & Gudas 2011). Therefore, the Object Management Group (OMG) initiated the creation of a domain-specific language and released the first version of SysML (OMG 2012a) in 2007. As SysML is a profile of UML, it has been easily adopted by most UML tool vendors.

It is commonly misunderstood in SE that SysML as a modeling language is enough to successfully apply MBSE in the organization. (Silingas & Butleris 2009) clearly state that the modeling language is just the language, and must be combined with a methodology to be useful. The number of methods available for MBSE is not significant (Nikolaidou *et al*. 2009). Analysis of MBSE methods and enterprise architecture frameworks reveals that the majority are conceptual and thus can hardly be used in combination with modeling languages, such as SysML, in practice. In contrast, the MBSE Grid approach proposed by (Morkevicius *et al.* 2017), (Mazeika *et al.* 2016) is fully compatible with SysML. Based on a transparent system architecture framework, it clearly defines the modeling process, reveals what model artefacts should be produced in each step of system lifecycle, and explains how to manage traceability relationships.

It is important to understand that there is no way to adopt MBSE without having a specific software tool or framework of integrated tools (Cloutier & Bone 2010), (Spangelo *et al.* 2012). The strength of MBSE relies on the tools. The market nowadays offers a broad selection of tools for systems modeling, each with its own strengths and weaknesses.

MBSE Adoption challenges

To successfully apply MBSE, organizations are obliged to implement proper practices, where language, method and framework, and tools are vital constructs. MBSE adoption in real world applications still struggles with huge challenges (Karban *et al.* 2011), (Albers & Zingel 2013), which neither the MB nor the SE part is able to handle. This is the main reason for the growth of interest in tools combined with methods even if users do not follow a standard modeling language. A set of challenges experienced by various MBSE adoption projects is listed and described is described in the following section:

Upfront Investment. MBSE adoption requires a substantial upfront investment, especially if it has not been considered before. This also includes determination of an effective investment strategy, accurate cost estimation and quantifying its return on investment.

Adoption Strategy. Two approaches dominate MBSE adoption: off-cycle (in a sandbox environment) or on-cycle (directly on productive projects). The first approach is considered ideal, as not all companies have the required budget and time. The second approach is much more challenging and introduces additional costs for running projects. Choosing the wrong strategy can negatively impact the benefits of MBSE.

Purpose and Scope Definition. A crucial basis for MBSE adoption is to define a clear purpose and scope (the *why* and *what*). Ideally, it must be precisely described before beginning the deployment. However, this is a challenge in real world applications, where modeling can be used in so many ways.

Awareness and Change Resistance. The human factor plays a central role, particularly if key players have different levels of MBSE knowledge and adequate time for training is not granted. Consequently, change is not always accepted, compared to existing approaches, it creates strong resistance due to the lack of expertise to deliver the required artefacts.

Executive Level Sponsorship. Although increased MBSE popularity has strengthened executive support, there are still conflicting MBSE adoption goals between short-term driven employees who care about low adoption cost, with others aiming at more adoption quality and long-term solutions.

Method Definition and Extension. It is often necessary to customize an appropriate method according to a defined purpose and scope. It is a challenge to set up the required method, document it and facilitate it with modeling rules, guidelines, tool customizations and training materials. Further challenges arise when new method extensions are needed.

Modularity and Reusability. Many organizations still follow an opportunistic and isolated reuse approach, where a set of data is copied and pasted from one context to another. Unfortunately, this still happens even with system models and results in losing the "*source of truth*" as soon as the copied source or pasted target is changed.

Complexity Management. The evolution of systems through the growing number of components, functions and interactions has dramatically increased their complexity. The issue here is with both the high number of model elements and the dependencies between the whole elements and model(s). Very often this complexity level takes existing methods and tools to the limit.

Tool Dependency and Integration. Companies need to pick a set of tools and train employees accordingly. Such a decision is not an easy task and there is no tool that satisfies all needs. Moreover, integration between systems modeling tools and others, such as simulation or requirements, is still solved with specific solutions.

Large Models Visualization. Different team members are involved in querying the model contents. Unfortunately, existing tools require additional training effort, and customizing the layout of model elements and diagrams is time consuming. Additional challenges appear in large models, where model navigation and understanding become highly complicated.

Related Works

(Friedenthal, Moore & Steiner 2014) describe a typical improvement process for deploying SysML as part of an MBSE approach. The process includes several steps which should be applied incrementally to improve the organization's capability. The steps start with "monitor and assess", and iteratively go over "plan the improvement", "define the changes", "pilot the approach" and "incrementally deploy the changes". The authors stress the need to plan, pilot and incrementally deploy, while considering the impacts on the systems engineering process, methods, tools, and training in order to achieve a successful SysML deployment.

An extensive application example is explained in (Parrott *et al.* 2016) through the implementation status of MBSE at NASA GRC from 2007 through 2016. The authors identified various MBSE adoption challenges, including the significant investment required to become an effective MBSE practitioner, the collaboration in a multi-center modeling effort and lack of resources. It was also noted that more experienced modelers often do not have the required availability and contracted modeling support can be expensive.

(Hallqvist & Larsson 2016) applied systems engineering principles at Saab to introduce MBSE into their organization. They reported the problems faced during the early phases of MBSE introduction and went through other problems in later phases. Interestingly, they highlight how fundamental systems engineering principles can help overcome MBSE introduction obstacles, by addressing all stakeholders during the change process, focusing on the purpose of the change and ensuring a proper communication plan and leadership presence.

Many best practices in the area of MBSE adoption are summarized in the ISO/IEC/IEEE 15288:2015 standard. This standard states that the process of systems engineering as defined can be decomposed into the phases of agreement, organizational project-enabling, technical management, and technical process. The international standard aligns with both document-based and model-based approaches for SE. However, it identifies no difference between them; the only difference recognized by the standard is the media to store technical project data. Consequently, the standard does not describe the transformation to MBSE, and many organizations find it challenging, as it requires revising and updating the entire process of SE. The D3 MBSE Adoption Toolbox proposed in this paper is intended to supplement ISO 15288.

Kongsberg Defence and Aerospace – Joint Strike Missile Project. A number of different technologies to be integrated, a lot of software (60% of the system requirements affects software), and 30-year lifecycle perfectly define the complexity of the next generation cruise missile project, a missile to be integrated on the F-35 Joint Strike Fighter. One of the initial objectives for systems engineers was to develop a system architecture understandable for maintenance, mid-life update, and new product variants while taking project complexity into account. To achieve these objectives, the systems engineers chose MBSE as a way forward. Another objective was set to establish a consistent System Architecture Model (SAM) (Soegaard 2016).

Adoption of MBSE was not an easy task. A new methodology had to be developed and learned. It took several years and 10 workshop and training sessions to mature the architecture framework. However, many people were and are now not motivated to use the modeling tool. The majority claim that SysML, though powerful and expressive, is too complex. The key success factors in the project included definition of a language subset and a strict guideline to develop large models, and establishment of a reference model expressing which subset of SysML to use for what purpose. Clear terminology is essential in communicating the model; however, it is a challenge to develop methodology and guidelines in parallel with product development (Soegaard 2016).

The project is considered a success. However, there are places to improve and evolve, such as model-based test and verification, MBSE experts training program and career ladder, product line engineering, etc. So far, a good foundation has been built. SAM expresses requirements, functional architecture, and logical architecture. There is smooth transition to software component design, including code generation for interfaces. Most importantly, there is commitment from management to move forward with MBSE (Soegaard 2016).

Bombardier Transportation (BT). The systems engineering challenges with respect to technology and personnel and the MBSE adoption at BT are presented in (Chami *et al.* 2015). The BT System Modeling Method (SysMM) describes how BT engineers analyze, define, and represent their system of interest using a Model-Based Systems Engineering approach. The purpose of the SysMM is to manage complexity and increase quality of the design artefacts to reduce development costs. The BT SysMM demonstrates the level of customization, big organizations with complex systems as rolling stock, and the need to spend and invest to begin implementing MBSE on real world projects. Investment includes the customization of modeling languages and tools together with the suitable method definition, providing training courses, parallel coaching and guidelines.

Additionally, adopting model-based solutions in other applications (e.g., testing, variant management), presents new challenges, especially if the big picture was not considered from the beginning. BT used a SysML-based variability management solution to define, analyze and configure the variability and commonality of different rolling stock product families (Chami *et al.* 2017). They concluded that variability configuration is only the first step towards Product Line Engineering (PLE) and a big picture solution should combine both PLE and MBSE together. Therefore, MBSE alone is not enough and it is crucial to integrate its adoption with other process activities. The authors in (Heinz *et al.* 2017) show how Scrum supports the deployment of MBSE on a rolling stock project through improving collaboration and team spirit by sharing small and regular successes, along with the possibility of a fast reaction for immediate visibility of any delay throughout the scrum sprints.

MBSE Grid. It is worth mentioning the MBSE grid method for modeling systems. Though it is not a complete MBSE deployment framework, it describes the best practices of organizing the work around the system model. The idea of the MBSE Grid approach originated after the demand for a simplified MBSE method was identified. The new approach summarizes the experience of numerous MBSE adoption projects in many organizations from various industry sectors, including KDA and BT. In contrast to other methods, the MBSE Grid approach is completely applicable in practice (Mazeika *et al.* 2016), because it is both fully compatible with SysML and it clearly defines the modeling process, which is based on the best practices of the systems engineering process.

The MBSE Grid approach is based on a framework, which can be represented as a Zachman style matrix (Zahman 1987). It is designed to guide system engineers through the modeling process by helping them answer questions, such as how to organize the model, what is the modeling workflow, what model artefacts should be produced in each step of that workflow, how these artefacts are linked together, etc. (Morkevicius et al. 2017), (Morkevicius & Jankevicius 2015).

This approach enables users to overcome certain MBSE adoption challenges introduced in the preceding section. It can be adopted fully or partially by using a selected subset of the framework or modifying it to conform to the engineering process and best practices of the organization. The concepts of definition and usage determined by the MBSE Grid approach enable the reusability and modularity of information. Traceability rules among different model artefacts defined by the approach help to manage the complexity and facilitate the change management. These advantages should help to overcome or minimize the cultural resistance of the new strategy for SE adoption.

Studies in the field find the same challenges as introduced in the preceding section. The problem is that they provide methods of dealing with various challenges in silos, which are only a partial solution to the overall problem expressed by this paper. However, all analyzed material provides very valuable input to the solution proposed in this paper.

D3 MBSE Adoption Toolbox

This section presents the D3 MBSE adoption Toolbox.

MBSE Adoption Components

Most organizational driven MBSE methodologies and frameworks, e.g., the BT SysMM (Chami *et al.* 2015) and KDA System Architecture Framework (Soegaard 2016), are defined as having three major components: language, method and tool. Compared to theoretical methods, where a modeling tool is often missing, consideration of these three components from the early phases of the MBSE adoption is crucial. The D3 MBSE adoption Toolbox goes beyond that to include a fourth component: personnel. The following figure shows the four components of the D3 MBSE adoption Toolbox and their interfaces.

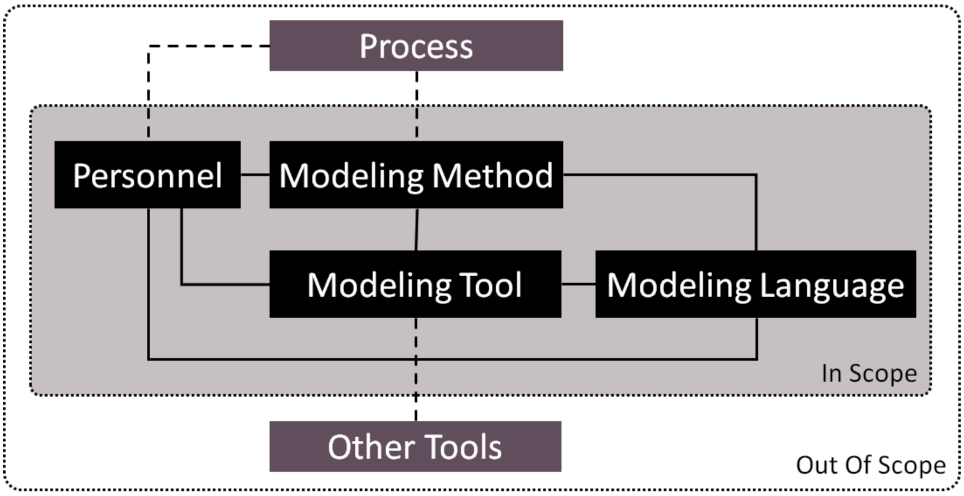

Figure 1. D3 MBSE Adoption Toolbox Components

Modeling Language. The de facto standard for MBSE is SysML. It is a critical enabler for MBSE. However, SysML is neither a framework nor a method: it provides no information about the modeling process and must be combined with some method to become truly applicable. Furthermore, SysML was meant to remain a general-purpose modeling language. Therefore, it is often necessary to extend SysML based on the defined method.

Modeling Method. There is a common misconception that the systems engineering process, e.g., ISO15288, solves the challenge of creating a system model from A to Z. Unfortunately, this is not the case. Moreover, it is not only the creation of models that matters. Users must also take model governance and model usage into account. (Morkevicius 2013) introduced a concept of modeling culture consisting of the following constructs: (i) model creation, (ii) model governance, and (iii) model usage. The latter two are addressed rarely in practice.

(Estefan 2008) presents a survey on MBSE methodologies and emphasizes that if a process defines the “what”, a modeling method should define the “how”. Although the number of MBSE methods is growing over time, none are 100% ready for use, nor are they compliant with any organization’s needs. Organizations are forced to invest time and money for the development of their own method(s).

Modeling Tool. There is no way to adopt MBSE without having specific software. The strength of MBSE relies on the choice of software. The market nowadays offers a broad selection of software tools for systems modeling, each with its own strengths and weaknesses. It becomes more and more obvious that a thorough tool study must be performed before choosing a tool (Friedland *et al.* 2017).

Personnel. The personnel component concerns employees within an organization who are engaged directly or indirectly with the MBSE adoption. They should involve both technical and management aspects of MBSE. Although the personnel component depends on the organization size, MBSE scope and goals, product complexity and other factors, its importance is often underestimated and mentioned only during the early stages of MBSE adoption planning.

MBSE Adoption Interfaces

Among MBSE adoption components. A modeling method must communicate with all in-scope components of MBSE adoption. It needs to be closely aligned to language (e.g., the OOSEM or MBSE Grid are both aligned with SysML) and based on a specific modeling tool. A tool-agnostic method is incomplete and more inefficient than the tool-specific method. Though some theorists claim that you can have multiple modeling tools, in practice you must have at least one within the scope of a project to address model governance and model usage practices. A single-tool environment can significantly reduce tool tailoring and training costs.

Tool selection depends on the language and method. It must support the language of choice and the method or at least be easily tailorable to support both. Tool usability and ergonomics are two other important factors for success. Personnel are not willing to work with a tool that is cumbersome and difficult to use. Method and language specific tool training is necessary in any case.

Based on our observations in different industry sectors, the key success of MBSE adoption lies in a clear consideration, coordination and supervision of the personnel involved and, most importantly, the correlation between them and other concepts of MBSE adoption (e.g., method, process, tool). A common point of agreement is that personnel are the drivers of the MBSE adoption. A successful MBSE adoption can only be accomplished with the right personnel setup and utilization. For example, a modeling method needs to be aligned with personnel skills, experience, knowledge, and cultural appeal. It must be clear for both directly involved engineers and other stakeholders in order to make the modeling process and model a truly valuable communication tool within the company. Furthermore, the inputs and outputs of any modeling methods must be consistent with the applied process mandates. Personnel must also be trained in the language to be able to use it as the communication tool, both internally and with stakeholders.

External. MBSE adoption concepts need to be aligned to a process, e.g., ISO/IEC/IEEE 15288:2015. This alignment is crucial to connect systems engineering practices to other disciplines (requirements engineering, software engineering, mechanical engineering, electronics engineering, etc.). The alignment needs to be implemented in two levels. First, in the process level, the modeling method steps that the SE processes support must be clearly indicated. Secondly, at the resources level, the required resource configuration must be defined, including both personnel and software tools to support every SE process.

The D3 Toolbox shown in Figure 2 supports both off-cycle (in a sandbox environment) and on-cycle (directly on productive projects) adoption strategies. Although the off-cycle environment is ideal, it still depends on the organization's upfront investment.

The following paragraph describes the three phases of the D3 Toolbox.

D3 MBSE Adoption Toolbox Phases

The D1 Phase: Definition. Although the starting point of the MBSE adoption is not easily identified, as defined by the D3 Toolbox, it triggers the 1st D, the Definition phase. Normally, it starts with small initiatives and spreads in different dimensions: (1) the application dimension, such as testing, verification and validation, variant management, document generation; (2) a system hierarchy dimension such as the system level itself – the delivered product – or a subsystem or set of system components; or (3) a personnel dimension to identify who will contribute to MBSE adoption and who will not. Regardless of the identification of a start of the MBSE adoption, the D3 Toolbox defines a starting point which triggers the 1^{st} D phase.

In this phase, a deep analysis on the topic needs to result in a clear decision as to whether or not MBSE will be adopted in a particular organization. If the decision is yes, the question "what?" must be considered. "What?" stands for the essentials, scope, goals and expected benefits of the MBSE adoption. Often, the definition phase is accelerated or shortened, and mixed with other phases due to pressure to meet project deadlines.

Further in this paper we list tasks included in the Definition phase, resulting in a MBSE development and deployment plan.

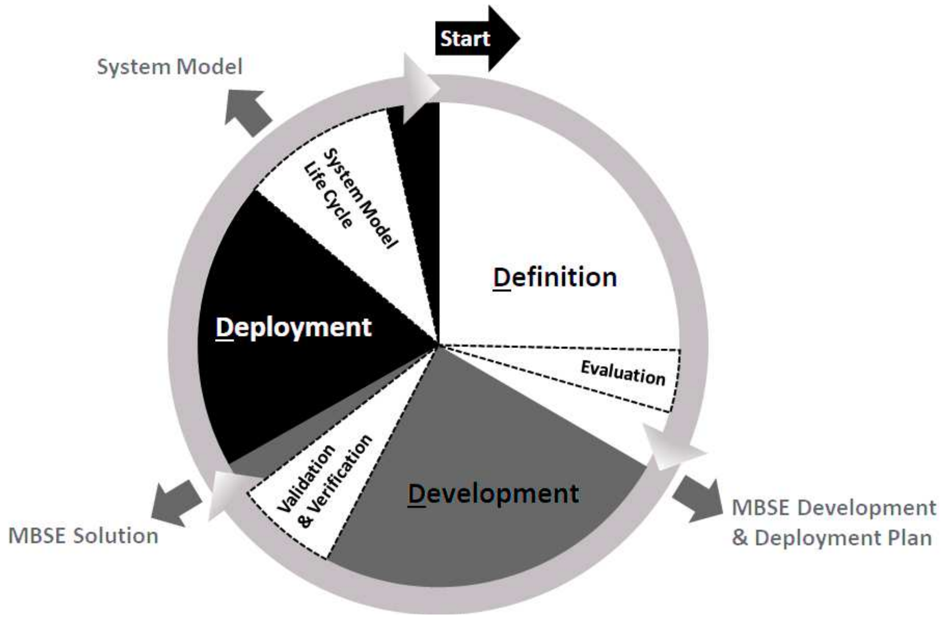

Figure 2. The D3 MBSE Adoption Toolbox Overview

Definition of MBSE adoption scope and purpose: Determining a suitable scope and purpose of MBSE adoption is very important, and should be accomplished as early in the adoption process as possible. While the benefits of MBSE are well described in the literature, each organization needs to identify the specific benefits that are driving them to move to a model-based approach. For most organizations, key benefits such as reducing development costs, increasing quality of the design artefacts, managing complexity, centralized information sharing, enhanced communication, reduced integration risk and increased reusability drive the assessment of return on investment (ROI) and act as the governing principles that make modeling beneficial. With respect to the definition of the scope, the task is to answer two questions: "what is included in the MBSE domain?" and "what is not?" These questions precisely define the boundaries of the modeling effort, and can include:

- Which requirements will be refined by models and which will not?
- Which functions will be described using models and which will be captured by plain text?
- Which system hierarchy level and interfaces will be covered by the scope of modeling and which will not?
- Which personnel will participate and who will not?

The definition of the scope could include other dimensions, such as integration with other processes and tools, collaboration with suppliers, sales, manufacturing, and in some specific instances a customer, if he demands models as the primary artefacts.

Establishment of MBSE team: A team of MBSE experts could be known by several names, such as "center of excellence" (CoE) or "center of competence" (CoC). In this paper we refer to the team as CoC. There is no difference between establishing a CoC and establishing other teams within the organization; however, not all organizations possess the budgets needed for a CoC. Since MBSE is still in an early stage of maturity (INCOSE 2014), the CoC is commonly a mixed team of both internal personnel and external consultants. The main challenge lies in finding the right personnel with the proper competencies. General MBSE and deep SysML experience are not enough to become a member of the CoC. It is also unreasonable to expect that a single person knows everything. Domain experts need to be involved to ensure high quality of the content of models. Project managers need to play the modeling governance role and align it with modeling goals. All team members should collaborate regularly toward the same goal. (Heinz et al. 2017) present an example of using scrum to boost MBSE deployment, while (Holt & Perry 2017) highlight the roles of people and their competencies by providing the meta model.

Awareness course: This task concerns achieving the same level of MBSE knowledge among the people involved in the Definition phase. At the end of this phase, both development and deployment plans should be clear for stakeholders. Therefore, the required awareness needs to be provided. It is not meant to make everybody a SysML expert, but it should address MBSE aspects at the correct level to achieve a common glossary, as well as an understanding of what the Definition phase is all about.

Problem statement and challenges: Although this task might sound theoretical for fast-result-oriented personnel, it is very important to consider it in the Definition phase. Modeling without addressing the motivation will result in useless models, which are normally discovered only in the MBSE Deployment phase (as shown in Figure 2). Thus, different problem statement methods need to be used (e.g., mechatronics concept design) to help define the related problems (normally known as systems engineering challenges). Moreover, problem statements differ across organizations based on various factors, such as product complexity, company size and number of stakeholders involved.

Return on Investment prognosis: The previous task of defining a problem statement should be coupled with the business criteria. A clear investment cost and expected benefits estimation should be implemented with a dedicated MBSE business case. These criteria should be measured regularly during the deployment phase through the modeling governance role to ensure immediate feedback, the ability to react quickly and to document lessons learned for future deployments.

Quality assurance and risk management: Risk results should tune the concepts of MBSE adoption. Additionally, identification and analysis of all existing processes which could be affected by introducing MBSE need to be performed, especially identification of process tasks, where the transition from documents to models would take place. Also, all inputs and outputs need to be listed to consider the relationships with other tasks. A respective evaluation criterion needs to be settled in order to measure quality factors.

Delivery of development and deployment plan: MBSE adoption needs to be planned and managed, just as any adoption of new technology or approach would be. This is crucial for MBSE adoption and must be done as quickly as possible. The lack of planning or managing effort could be difficult to recognize if the required MBSE knowledge and quality assurance is not available. Therefore, there is a need to carry out a business plan and business case in parallel to all technical MBSE related issues.

It is vital to ensure the monitoring, transparency and governance are in place for a successful MBSE adoption. All related information should be collected prior to starting the Development and Deployment phases of MBSE adoption.

The D2 Phase: Development. The development phase typically starts in parallel or shortly after the Definition phase is completed. The ideal D3 Toolbox scenario suggests that the 2nd D starts only after the delivery of MBSE Development & Deployment Plan. However, in practice, this will not be the case if the MBSE competence does not exist. In this phase, the development of the MBSE adoption should take place. Development means developing the MBSE approach, not the model of the system or service. Additionally, a set of the most important concepts needed to have the MBSE solution ready to be deployed on projects is defined.

Modeling ontology and glossary: SysML is considered to be a critical enabler for MBSE. It provides a list of generic MBSE concepts to be used in the scope of MBSE development, as well as supplying a versatile extension mechanism to adapt language to meet specific industry requirements. However, in defining the final ontology to be used, it is important not only to extend, but also to shrink the number of concepts provided by SysML.

Model libraries: MBSE enables the reuse of models. It is a common practice to define libraries for elements that are going to be reused throughout a number of different projects, e.g., interface libraries, component libraries, unit libraries, etc. If these libraries are already defined in other projects, they can be reused instead of redefined.

Method guidelines: The Guideline component is an essential component of MBSE development. It binds MBSE language, MBSE method, MBSE tool and Personnel together. It significantly reduces the learning curve for new employees, as well as existing employees who are not familiar with MBSE. A good practice is to have an example model illustrating the use of MBSE within an organization.

Customized training courses: It is very important to understand that existing training courses provide the basis to start with modeling tasks. It is easy to be trained in SysML; however, method-related information is not part of SysML and usually varies among different organizations. Therefore, there is no way to avoid defining customized training material according to the MBSE adoption strategy and delivering a customized training course to personnel.

Model checking rules: One method to ensure the quality of models is to implement rules that check model consistency and completeness. Validation can be carried out manually via model reviews, as well as automated with the dedicated validation rules implemented in the modeling tool.

Tool customizations: Several tools exist for MBSE. Although they share the same set of features, each has its own specifics. The advanced adoption of MBSE requires several tool-related aspects, such as language extensions, validation rules, automated scripts, etc. These should be aligned with modeling method and organization needs. The main challenge occurs while integrating the MBSE tool with the rest of the engineering toolchain (e.g., requirements management, testing, simulation).

MBSE solution delivery: All the above tasks form the development of the MBSE solution. It is presented here as a delivery of a product because of its importance. Its verification and validation are also vital to ensure a successful deployment phase.

The D3 Phase: Deployment. Typically, this phase takes place in every MBSE adoption case. It is all about the implementation of a system model. The main concern here is ensuring that implemented system model reflects the real-world system and achieves the MBSE purposes set at the beginning of the MBSE adoption. In practice, one can deduce this by simply asking the question: “Is the system model still being used?” If the answer is yes, then the model lifecycle is still active and the model is still alive. The lifecycle definitely has an end, but the early end of a model lifecycle indicates that something went wrong during the deployment phase. Unfortunately, in MBSE, there are few cases where maintaining a system model takes more effort than starting with a new one from scratch (with the transformation of the technical content of the old one).

One of the main pitfalls that occur during the deployment phase is when a new purpose is introduced or a modification of the scope is requested. For instance, this may happen when a team starts deploying the MBSE approach for requirements engineering and functional architecture and a request comes to include testing (e.g., create automatic test scripts for the functional architecture). Although the benefits of such a request might sound reasonable, the risk is barely manageable if the modifications on the actual models are not fully considered. In summary, during the deployment phase the team must consider that the delivery of valuable models, with longer lifecycles, is the key to success.

Deployment Governance: Although governance relies mainly on the personnel factor and is difficult to achieve, a certain level of governance is necessary during the Deployment phase. Here, governance includes several aspects, such as monitoring the modeling status with respect to the time line, measuring the model quality through the necessary reviews, ensuring the participation of all stakeholders, and, most importantly, reacting immediately to prevent any risk of delay. On the other side, too much governance is not appreciated. Therefore, a balanced governance combined with transparency leads to the best results.

Quality of valuable models: The list of quality criteria could get very long. It is also organization specific. The literature already describes a common set of criteria to ensure valuable models. The first example of quality criteria is abstraction levels: for complex systems with a large number of model elements, there is a need to define abstraction levels to deal with complexity. These levels can correspond to the system hierarchy levels or can be based on the method tasks, such as operational, functional or technical analysis, as presented in (Chami *et al.* 2015), for operational, functional, and technical levels. The second example relates to documentation: the documentation of model elements enriches the system model quality and improves the understanding among different stakeholders (e.g. requirements, functions, safety).

System model delivery and lifecycle: This task is related to the “definition of done”. The system model should adhere to the MBSE adoption purposes and deliver its targets. Moreover, the less effort required for the maintenance, the better the configuration.

Application Cases

The D3 Toolbox depicted in Figure 2 shows an ideal case where all three MBSE adoption phases are equally split. In reality it is different. In this section, we summarize two different application scenarios with different characteristics based on two industry sectors. Due to non-disclosure agreements with industry partners, neither the names nor any technical details are mentioned in this paper.

Application Case 1. This case demonstrates MBSE adoption by a relatively small-size organization. The Definition phase, compared to the ideal version, is quite short. The MBSE team decided to consider a small scope and clear purpose. With respect to the scope, it spreads among various factors. The team focuses on two applications: one for requirements engineering and the other for functional architecture. Only functional requirements that are safety-critical are covered. The system hierarchy level is set to consider the system-of-interest level and its subsystems (components are not in the scope). The purpose mainly covers the transition of describing the system functions directly in the system model and not in documents. The system model in a dedicated modeling tool is the primary artefact and documents are generated directly from it. Other purposes were also considered, including improving quality, reducing recurring costs and speeding up the design and development period.

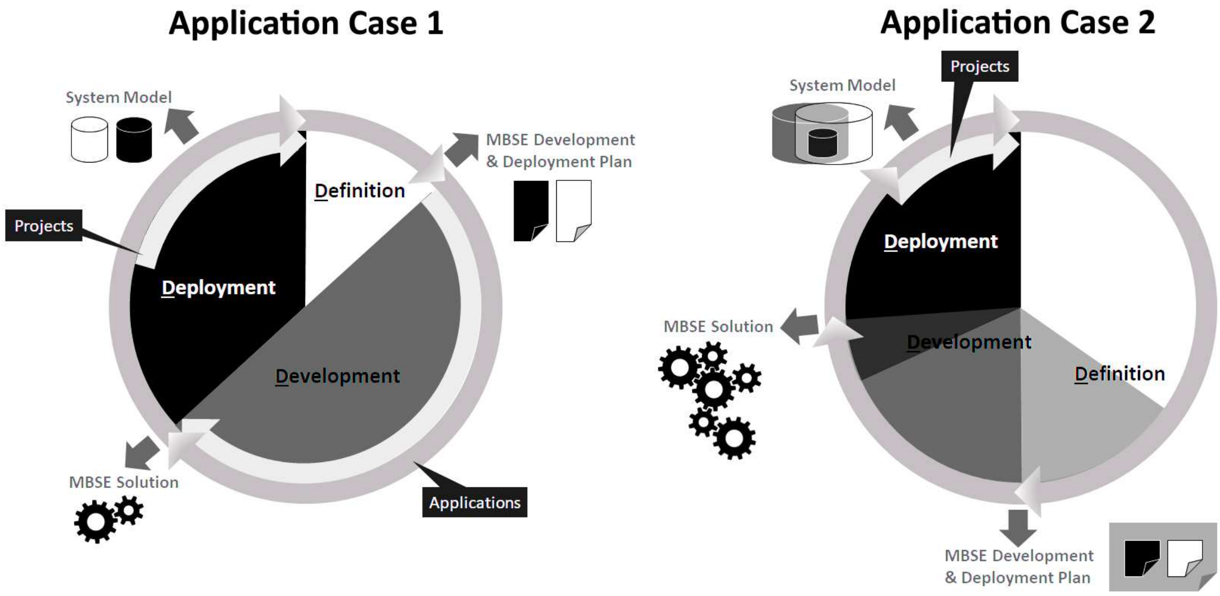

Figure 3. D3 MBSE Toolbox: Two Different Application Cases

The Definition phase generated MBSE Development and Deployment plans with all related information about resources, planning, and other business aspects. When the Development phase began with the implementation of the MBSE solution, it took some time because the organization had done this before and decided to invest in training internal employees rather than relying completely on external consultants. The Development phase of the Application Case 1 in Figure 3 shows two applications. Both were running in parallel to implement all needed factors, such as training, methods, guidelines and the clear schema for linking the requirements with the system model elements. A small effort was spent on creating a validation concept to ensure the quality of the information generated in documents.

The Deployment phase started directly on a project after the team was trained. After a period of time, the second project started adopting MBSE in parallel. A simple process was set to divide the team involved into different groups based on the functions each team covered. Each functional group included a modeler, requirements manager and functional architect. Reviews were only accepted by the functional architect after coordinating the exported documents with model content (diagrams and textual description).

Application Case 2. This case demonstrates an MBSE adoption by a large-scale organization geographically spread over different sites. The D3 MBSE Adoption Toolbox phases overlap because the company started MBSE adoption on more than one application in parallel. The MBSE team decided to create a core network of MBSE key users to ensure reusability and proper collaboration. With respect to scope, both systems engineers and domain-specific engineers were involved (i.e., simulation, testing and safety). Therefore, six model-based applications were considered: requirements engineering, functional architecture, testing, simulation, safety and variant management. The team also decided to consider all their system hierarchy levels (system, subsystems and components). One of the main goals was to adopt MBSE directly from the beginning on a platform before doing it on the application project. This was requested due to the urgent need to reduce recurrent cost and ensure the usage of standardized design assets rather than reinventing it for every project. Therefore, the MBSE Development and Deployment plan highlighted the business aspect (included sales engineers and projects managers), not just the technical one.

The Development phase began before finishing the Definition phase. Training courses were split into two parts: generic related topics (e.g., Fault Trees, variant modeling), provided by consultants, and specific topics (method, tool customizations, models integration), customized and trained by internal personnel. Moreover, the development phase included many aspects as opposed to Application Case 1, such as model libraries, model validation on different levels, automated wizards to help in creating model elements faster, web reporting of system models and with capturing integrated comments.

The preparation for the Deployment started before the Development phase ended, due to the necessity of testing the large-scale MBSE solution and various limitations in the tools, including lack of specifying what was really needed. Achieving a standardization level for full usability required a deep dive in all the deployment steps. The team decided to run all development results on two different examples as a pilot to ensure the clarity and quality of the final MBSE solution. Therefore, the official start of the Deployment phase began with the implementation of the generic model. A generic model for the platform was created and imported into two application projects. Based on a clear analysis and definition of generic and specific content, modelers placed their content in the appropriate system models.

Model governance was crucial in this application case. The size of the generic and specific system model elements, the team involved, issues reported and the complexity of the system of interest reflected in a huge delay during deployment. Therefore, the team enhanced the Deployment phase with the application of a scrum approach, where collaboration and immediate action were much improved.

Conclusion and Future Work

This paper collects and represents all concepts organizations must be aware of when adopting MBSE and a process of MBSE adoption in the form of a D3 Toolbox. Two application cases revealed that the D3 Toolbox will not be applied 100% as it is defined. However, it gives a clear understanding of what needs to be considered in the MBSE adoption process.

Although the D3 Toolbox is still in its very early stages, it has the potential to help organizations, especially novices to MBSE, to successfully adopt this new evolving technology.

In the future, we aim to optimize the D3 Toolbox for different modeling applications, e.g. functional analysis, testing, and variability. In particular, we will highlight where the differences exist among different applications. Additionally, our vision goes further to consider the interfaces between the phases of the different modeling applications (e.g., the interface between the definition phase of requirements application and other phases of the functional architecture application). Our next step is to describe the fulfillment of the D3 toolbox components in relation to the MBSE adoption challenges described in this paper.

References

Albers, A. & Zingel, C., 2013, ‘Challenges of Model-Based Systems Engineering: A Study towards Unified Term Understanding and the State of Usage of SysML’, in Abramovici, M. & Stark, R. (eds), Smart Product Engineering. Lecture Notes in Production Engineering, pp. 83–92, Springer, Berlin, Heidelberg.

Chami, M., Forlingieri, M. & Oggier, P., 2017, ‘Model-Based Variability Management Solution with SysML’, in Prostep ivip, Produkt Daten Journal Nr. 1 2017, pp. 38-41.

Chami, M., Oggier, P., Naas, O. & Heinz, M., 2015, ‘Real World Application of MBSE at Bombardier Transportation’, presented at SWISSED 2015, viewed 10 November 2017, from https://goo.gl/ADEtmh.

Cloutier, R. & Bone, M., 2010, ‘Compilation of SysML RFI- Final Report, Systems Modeling Language (SysML)’.
Delp, C., Lam, D., Fosse, E. & Cin-Young Lee, 2013, ‘Model based document and report generation for systems engineering’, presented at Aerospace Conference, IEEE.
Estefan, J. A, 2008, *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. INCOSE MBSE Focus Group.
Friedenthal, S., Moore, A. & Steiner, R., 2014, *A Practical Guide to SysML - The Systems Modeling Language*, MK/OMG Press, Morgan Kaufmann.
Friedland, B., Herrold, J., Ferguson, G., & Malone, R., 2017, ‘Conducting a Model Based Systems Engineering Tool Trade Study Using a Systems Engineering Approach’, in INCOSE International Symposium 27 proceedings, pp. 1087-1099.
Hallqvist, J. & Larsson, J., 2016, ‘Introducing MBSE by using Systems Engineering Principles’, in INCOSE International Symposium 26 proceedings, pp. 512–525.
Heinz, M., Pennisi, A., Parinaz, E.D., Oggier, P. & Naas. O., 2017, ‘Modeling “Agile” at Bombardier Transportation’, presented at SWISSED, 4th September, 2017.
Holt, J. & Perry, S., 2017, *Don’t Panic! The Absolute Beginner’s Guide to Model-Based Systems Engineering.* INCOSE UK Ltd.
INCOSE, 2014. ‘A World in Motion: Systems Engineering Vision 2025’. International Council on Systems Engineering (INCOSE), 2014.
International Council on Systems Engineering (INCOSE), 2007, *Systems Engineering Vision 2020*, (version 2.03).
ISO/IEC/IEEE, 2015, *Systems and software engineering - System life cycle processes*, (15288:2015).
Karban, R., Weilkiens, T., Hauber, R., Zamparelli, M., Diekmann, M. & Hein, A. M., 2011, *Cookbook for MBSE with SysML*, Tech. Rep.
Mazeika, D., Morkevicius, A. & Aleksandraviciene, A., 2016, ‘MBSE driven approach for defining problem domain’, in 11th System of Systems Engineering Conference (SoSE), Kongsberg, pp.1-6.
Morkevicius, A., 2013, ‘How to build a sustainable modeling culture’, presented at Nordic Systems Engineering Tour: Hamburg (April 15), Copenhagen (April 16), Stockholm (April 17).
Morkevicius, A., Aleksandraviciene, A., Mazeika, D., Bisikirskiene, L. & Strolia, Z., 2017, ‘MBSE Grid: A Simplified SysML-Based Approach for Modeling Complex Systems’, in INCOSE International Symposium 27 proceedings, pp. 136-150.
Morkevicius, A. & Gudas, S., 2011, ‘Enterprise Knowledge Based Software Requirements Elicitation’, in Information Technology and Control vol. 40, no. 3, p. 181-190.
Morkevicius, A. & Jankevicius, N., 2015, ‘An approach: SysML-based automated requirements verification’, in Systems Engineering (ISSE), 2015 IEEE International Symposium, pp. 92-97.
Nikolaidou, M., Tsadimas, A., Alexopoulou, N. & Anagnostopoulos, D., 2009, ‘Architecture Framework to Systematically Perform Model-Based System Engineering Activities’, in 42nd Hawaii International Conference proceedings.
Object Management Group (OMG), 2003, *MDA Guide*, (Version 1.0.1).
Object Management Group (OMG), 2012a, *Systems Modeling Language (OMG SysML)*, (Version 1.3).
Object Management Group (OMG), 2007, *Unified Modeling Language (OMG UML) Infrastructure*, (Version 2.1.2), Needham.
Parrott, E., Trase, K., Green, R., Varga, D. & Powell, J., 2016, ‘NASA GRC MBSE Implementation Status’, viewed 10 November 2017, from https://goo.gl/M95SVc.
Silingas, D. & Butleris, R., 2009, ‘Towards customizing UML tools for enterprise architecture modeling’, in IADIS international conference proceedings.

Soegaard S. E. 2016 Adopting MBSE using SysML in System Development – Joint Strike Missile (JSM), No Magic World Symposium 2016. from https://vimeopro.com/user28256466/no-magic-world-symposium-2016-presentations/page/3

Spangelo, S. C. *et al*., 2012, 'Applying Model Based Systems Engineering (MBSE) to a standard CubeSat', IEEE Aerospace Conference, Big Sky, MT, pp. 1-20.

Zachman, J. A., 1987, 'A framework for information systems architecture', IBM Syst. J., vol. 26, no. 3, pp. 276–292.

Biography

Mohammad Chami is a MBSE expert with a solid academic and industrial experience in modelling languages, processes, developing and deploying methods for systems modelling and customizing its tools. Currently, he is a managing consultant at IBM Global Business Services, leading significant client engagements in MBSE, Watson IoT and Cognitive Engineering. Prior to joining IBM, he was employed as a modeling expert at Bombardier Transportation, with a primary focus on the development and deployment of MBSE on operation projects across all BT divisions, leading the MBSE key users' network and frequently giving MBSE training courses. He holds two master's degrees in Electronics and Mechatronics, the OMG Certified Systems Modelling Professional Certificate (OCSMP) and is currently pursuing his PhD at Toulouse University.

Aurelijus Morkevicius is an OMG Certified UML, Systems Modeling and BPM professional. Currently, he is the Head of Solutions Department at No Magic Europe. He is experienced with model-based systems and software engineering, as well as defense architectures (DoDAF, NAF). Aurelijus works with companies such as BAE Systems, Bombardier Transportation, Deutsche Bahn, ZF, Ford, SIEMENS, and BMW. He is also the co-chairman and one of the leading architects for the current OMG UAF standard development group. In addition, Aurelijus is actively involved in educational activities. He received a PhD in Informatics Engineering from Kaunas University of Technology in 2013. Aurelijus is also a lecturer, author of multiple articles, and a speaker at multiple conferences.

Aiste Aleksandraviciene holds a Master of Information degree in Systems Engineering from Kaunas University of Technology and is an OMG certified systems modeling professional (OCSMP). Currently, she is a Solution Architect at No Magic Europe and is responsible for producing training material, organizing webinars, writing papers and making presentations at systems engineering community events to promote the MBSE culture. Her expertise area is model-based systems engineering with a special focus on managing system requirements.

Jean-Michel Bruel is head of the SM@RT team of the IRIT CNRS laboratory. His research areas include development of software-intensive Cyber-Physical Systems, methods/model/language integration, with a main focus on requirements and Model-Based Systems Engineering. He defended his "Habilitation à Diriger des Recherches" in December 2006 and obtained a full professor position at the University of Toulouse in 2008. He was head of the Computer Science department of the Technical Institute of Blagnac from 2009 to 2012.